Intel® Celeron® D Processor 300Δ Sequence

Specification Update

April 19th 2006

Notice: The Intel® Celeron® D Processor on 90 nm Process and in the 478-Pin Package and the Intel® Celeron® D Processor on 90 nm Process and in the 775 Land Package may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are documented in this Specification Update.

Document Number: 302354-018

Contents

§

Revision History

Version	Description	Date of Revision
-001	• Initial Release	June 2004
-002	• Added Errata L31-L35	August 2004
-003	• Added Errata L36-50	September 2004
-004	• Added E step information. Added Erratum L49-L52. Minor updates to Summary of Errata Tables. Updated Processor Identification Table.	September 2004
-005	• Added Erratum L53-L57.	October 2004
-006	• Added Erratum L58. Updated Processor Identification Table.	November 2004
-007	• Added Specification Clarification L1. Updated Errata L29 & L33. Updated Processor Identification Table.	April 2005
-008	• Added Specification Change L1. Updated Erratum L31 and L36. Updated Processor Identification Table.	June 2005
-009	• Added Erratum L59-72. Updated Processor Identification Table. Updated processor marking information.	June 2005 Out Of Cycle
-010	• Added Erratum L73.	July 2005
-011	• Removed Erratum L18 and replaced with new Erratum. Updated processor identification information.	Aug 2005
-012	• Removed Erratum L14 and L33 and replaced with new Erratum.	Sept 2005
-013	• Added Erratum L74 and L75.	Oct 2005
-014	• Updated Errata L25 in the Summary of Errata Table	Nov 2005
-015	• Updated Summary table of changes with G1 information. Added Erratum L76. Added Specification Clarification L1 and L2. Updated processor identification information.	Dec 2005
-016	• Launch of 355 processor. Updated processor identification information.	Dec 2005 Out Of Cycle
-017	• Added Erratum L77. Correct Erratum Number in Revision history table under version 15.	March 2006
-018	• Added Erratum L78 and L79. Updated processor identification information.	April 2006

§

Preface

This document is an update to the specifications contained in the documents listed in the following Affected Documents/Related Documents table. It is a compilation of device and document errata and specification clarifications and changes, and is intended for hardware system manufacturers and for software developers of applications, operating system, and tools.

Information types defined in the Nomenclature section of this document are consolidated into this update document and are no longer published in other documents. This document may also contain information that has not been previously published.

Affected Documents

Document Title	Document Number
Intel® Celeron® D Processor 3xx Sequence Datasheet - On 90 nm Process in the 478-pin Package	302253-005
Intel® Celeron® D Processor 3xx Sequence Datasheet – On 90 nm Process in the 775-Land Package	304092-005

Related Documents

Document Title	Document Location
IA-32 Intel® Architecture Software Developer's Manual Volume 1: Basic Architecture	http://developer.intel.com/design/pentium4/manuals/index_new.htm
IA-32 Intel® Architecture Software Developer's Manual Volume 2A: Instruction Set Reference Manual A–M	
IA-32 Intel® Architecture Software Developer's Manual Volume 2B: Instruction Set Reference Manual, N–Z	
IA-32 Intel® Architecture Software Developer's Manual Volume 3A: System Programming Guide	
IA-32 Intel® Architecture Software Developer's Manual Volume 3B: System Programming Guide	
IA-32 Intel® Architecture Software Developer's Manual Volume 1: Basic Architecture	

§

Nomenclature

S-Spec Number is a five-digit code used to identify products. Products are differentiated by their unique characteristics (e.g., core speed, L2 cache size, package type, etc.) as described in the processor identification information table. Care should be taken to read all notes associated with each S-Spec number

Errata are design defects or errors. Errata may cause the Intel® Celeron® D Processor on 90 nm Process and in the 478-Pin Package and the Intel® Celeron® D Processor on 90 nm Process and in the 775 Land Package behavior to deviate from published specifications. Hardware and software designed to be used with any given stepping must assume that all errata documented for that stepping are present on all devices.

Specification Changes are modifications to the current published specifications. These changes will be incorporated in the next release of the specifications.

Specification Clarifications describe a specification in greater detail or further highlight a specification's impact to a complex design situation. These clarifications will be incorporated in the next release of the specifications.

Summary Tables of Changes

The following table indicates the Specification Changes, Errata, Specification Clarifications that apply to the listed component steppings. Intel intends to fix some of the errata in a future stepping of the component, and to account for the other outstanding issues through documentation or Specification Changes as noted. This table uses the following notations:

Codes Used in Summary Table

Stepping

X:	Erratum, Specification Change or Clarification that applies to this stepping.
(No mark) or (Blank Box):	This erratum is fixed in listed stepping or specification change does not apply to listed stepping.

Status

PlanFix:	This erratum may be fixed in a future stepping of the product.
Fixed:	This erratum has been previously fixed.
NoFix:	There are no plans to fix this erratum.
Shaded:	This item is either new or modified from the previous version of the document.

Note: Each Specification Update item is prefixed with a capital letter to distinguish the product. The key below details the letters that are used in Intel's microprocessor Specification Updates:

A = Intel® Pentium® II processor
B = Mobile Intel® Pentium® II processor
C = Intel® Celeron® processor
D = Intel® Pentium® II Xeon® processor
E = Intel® Pentium® III processor
F = Intel® Pentium® processor Extreme Edition
G = Intel® Pentium® III Xeon® processor
H = Mobile Intel® Celeron® processor at 466 MHz, 433 MHz, 400 MHz, 366 MHz, 333 MHz, 300 MHz, and 266 MHz
K = Mobile Intel® Pentium® III Processor – M
L = Intel® Celeron® D processor
M = Mobile Intel® Celeron® processor
N = Intel® Pentium® 4 processor
O = Intel® Xeon® processor MP
P = Intel® Xeon® processor
Q = Mobile Intel® Pentium® 4 processor supporting Hyper-Threading Technology on 90-nm process technology
R = Intel® Pentium® 4 processor on 90 nm process
S = 64-bit Intel® Xeon® Processor with 800 MHz system bus
T = Mobile Intel® Pentium® 4 processor – M
U = 64-bit Intel® Xeon® processor MP with up to 8MB L3 Cache
V = Mobile Intel® Celeron® processor on 0.13 Micron Process in Micro-FCPGA Package
W = Intel® Celeron® M processor
X = Intel® Pentium® M processor on 90 nm process with 2-MB L2 cache
Y = Intel® Pentium® M processor
Z = Mobile Intel® Pentium® 4 processor with 533 MHz system bus

NO.	C0	D0	E0	LE0	G1	LG1	Plans	ERRATA
L1	X	X	X	X	X	X	No Fix	Transaction is not retried after BINIT#
L2	X	X	X	X	X	X	No Fix	Invalid opcode 0FFFH requires a ModRM byte
L3	X	X	X	X	X	X	No Fix	Processor may hang due to Speculative Page Walks to Non-Existent System Memory
L4	X	X	X	X	X	X	No Fix	Memory type of the load lock different from its corresponding store unlock
L5	X	X	X	X	X	X	No Fix	Machine check architecture error reporting and recovery may not work as expected
L6	X	X	X	X	X	X	No Fix	Debug mechanisms may not function as expected
L7	X	X	X	X	X	X	No Fix	Cascading of performance counters does not work correctly when forced overflow is enabled
L8	X	X	X	X	X	X	No Fix	EMON event counting of X87 loads may not work as expected
L9	X	X	X	X	X	X	No Fix	System bus interrupt messages without data and which receive a Hard Failure response may hang the processor
L10	X	X	X	X	X	X	No Fix	Processor flags #PF instead of #AC on an unlocked CMPXCHG8B instruction

NO.	C0	D0	E0	LE0	G1	LG1	Plans	ERRATA
L11	X	X	X	X	X	X	No Fix	FSW may not be completely restored after page fault on FRSTOR or FLDENV instructions
L12	X	X	X	X	X	X	No Fix	Processor Issues Inconsistent Transaction Size Attributes for Locked Operation
L13	X	X	X	X	X	X	No Fix	When the processor is in the System Management Mode (SMM), Debug Registers may be fully writeable
L14	X	X	X	X	X	X	No Fix	The Processor May Issue Front Side Bus Transactions up to 6 Clocks after RESET# is Asserted
L15	X	X	X	X	X	X	No Fix	Processor may hang under certain frequencies and 12.5% STPCLK# duty cycle
L16	X	X	X	X	X	X	No Fix	System may hang if a fatal cache error causes Bus Write Line (BWL) transaction to occur to the same cache line address as an outstanding Bus Read Line (BRL) or Bus Read-Invalidate Line (BRIL)
L17	X	X	X	X	X	X	No Fix	A write to APIC Task Priority Register (TPR) that lowers priority may seem to have not occurred
L18	X	X	X	X	X	X	No Fix	At Core-to-bus Ratios of 16:1 and Above Defer Reply Transactions with Non-zero REQb Values, May Cause a Front Side Bus Stall
L19	X	X	X	X	X	X	No Fix	Parity Error in the L1 Cache may cause the processor to hang
L20	X						Fixed	BPM4# Signal Not Being Asserted According to Specification
L21	X	X					Fixed	A 16-bit Address Wrap Resulting from a Near Branch (Jump or Call) May Cause an Incorrect Address to Be Reported to the #GP Exception Handler
L22	X	X	X	X	X	X	No Fix	Locks and SMC Detection May Cause the Processor to Temporarily Hang
L23	X						Fixed	PWRGOOD and TAP Signals Maximum Input Hysteresis Higher Than Specified
L24	X						Fixed	Some Front Side Bus I/O Specifications are not Met
L25	X						Fixed	Incorrect Physical Address Size Returned by CPUID Instruction
L26	X	X	X	X	X	X	No Fix	Incorrect Debug Exception (#DB) May Occur When a Data Breakpoint is set on an FP Instruction
L27	X	X	X	X	X	X	No Fix	xAPIC May Not Report Some Illegal Vector Errors
L28	X	X	X	X	X	X	No Fix	Memory Aliasing of Pages as Uncacheable Memory Type and Write Back (WB) May Hang the System
L29	X	X	X	X	X	X	No Fix	Interactions Between the Instruction Translation Lookaside Buffer (ITLB) and the Instruction Streaming Buffer May Cause Unpredictable Software Behavior
L30	X						Fixed	Changes to CR3 Register do not Fence Pending Instruction Page Walks
L31	X						Fixed	The State of the Resume Flag (RF Flag) in a Task-State Segment (TSS) May be Incorrect
L32	X	X	X	X	X	X	No Fix	Processor Provides a 4-Byte Store Unlock After an 8-Byte Load Lock

NO.	C0	D0	E0	LE0	G1	LG1	Plans	ERRATA
L33	X	X	X	X	X	X	No Fix	Front Side Bus Machine Checks May be Reported as a Result of On-Going Transactions during Warm Reset.
L34	X						Fixed	CPUID Instruction May Report Incorrect L2 Associativity in Leaf 0x80000006
L35		X	X	X	X	X	Plan Fix	Execution of IRET or INTn Instructions May Cause Unexpected System Behavior
L36	X						Fixed	The FP_ASSIST EMON Event May Return an Incorrect Count
L37	X	X	X	X	X	X	No Fix	Machine Check Exceptions May not Update Last-Exception Record MSRs (LERs)
L38	X	X	X	X	X	X	No Fix	MOV CR3 Performs Incorrect Reserved Bit Checking When in PAE Paging
L39	X	X	X	X	X	X	No Fix	Stores to Page Tables May Not Be Visible to Pagewalks for Subsequent Loads Without Serializing or Invalidating the Page Table Entry
L40	X	X	X	X	X	X	No Fix	Data Breakpoints on the High Half of a Floating Point Line Split may not be Captured
L41	X	X					Fixed	A Split Store Memory Access May Miss a Data Breakpoint
L42		X					Fixed	EFLAGS.RF May be Incorrectly Set After an IRET Instruction
L43	X						Fixed	Read for Ownership and Simultaneous Fetch May Cause the Processor to Hang
L44		X					Fixed	Writing the Echo TPR Disable Bit in IA32_MISC_ENABLE May Cause a #GP Fault
L45	X						Fixed	Cache Lock with Simultaneous Invalidate External Snoop and SMC Check May Cause the Processor to Hang
L46	X						Fixed	IRET Instruction Performing Task Switch May Not Serialize the Processor Execution
L47	X	X					Fixed	Incorrect Access Controls to MSR_LASTBRANCH_0_FROM_LIP MSR Registers
L48		X	X	X			Fixed	Recursive Page Walks May Cause a System Hang
L49				X			Fixed	When the Execute Disable Bit Function is Enabled a Page-fault in a Mispredicted Branch May Result in a Page-fault Exception
L50				X			Fixed	Execute Disable Bit Set with AD Assist Will Cause Livelock
L51				X			Fixed	The Execute Disable Bit Fault May be Reported Before Other Types of Page Fault When Both Occur
L52				X			Fixed	Execute Disable Mode may Cause Livelock
L53	X	X	X	X	X	X	No Fix	Checking of Page Table Base Address May Not Match the Address Bit Width Supported by the Platform
L54	X	X	X	X	X	X	No Fix	The IA32_MCi_STATUS MSR May Improperly Indicate that Additional MCA Information Has Been Captured
L55	X	X	X	X	X	X	No Fix	With TF (Trap Flag) Asserted, FP Instruction That Triggers an Unmasked FP Exception May Take Single Step Trap Before Retirement of Instruction
L56	X	X					Fixed	MCA Corrected Memory Hierarchy Error Counter May Not Increment Correctly

NO.	C0	D0	E0	LE0	G1	LG1	Plans	ERRATA
L57	X	X	X	X	X	X	No Fix	BTS(Branch Trace Store) and PEBS(Precise Event Based Sampling) May Update Memory outside the BTS/PEBS Buffer
L58	X	X	X	X	X	X	No Fix	Memory Ordering Failure May Occur with Snoop Filtering Third Party Agents after Issuing and Completing a BWIL (Bus Write Invalidate Line) or BLW (Bus Locked Write) Transaction
L59	X	X	X	X	X	X	No Fix	Control Register 2 (CR2) Can be Updated during a REP MOVS/STOS Instruction with Fast Strings Enabled
L60		X	X	X	X	X	Plan Fix	TPR (Task Priority Register) Updates during Voltage Transitions of Power Management Events May Cause a System Hang
L61				X		X	Plan Fix	VERR/VERW Instructions May Cause #GP Fault when Descriptor is in Non-canonical Space
L62				X			Fixed	The Base of a Null Segment May be Non-zero on a Processor Supporting Intel® Extended Memory 64 Technology (Intel® EM64T)Φ
L63				X			Fixed	Upper 32 Bits of FS/GS with Null Base May not get Cleared in Virtual-8086 Mode on Processors with Intel® Extended Memory 64 Technology (Intel® EM64T) Enabled
L64				X		X	No Fix	Processor May Fault when the Upper 8 Bytes of Segment Selector is Loaded From a Far Jump Through a Call Gate via the Local Descriptor Table
L65				X		X	No Fix	Loading a Stack Segment with a Selector that References a Non-canonical Address can Lead to a #SS Fault on a Processor Supporting Intel® Extended Memory 64 Technology (Intel® EM64T)
L66				X		X	No Fix	FXRSTOR May Not Restore Non-canonical Effective Addresses on Processors with Intel® Extended Memory 64 Technology (Intel® EM64T) Enabled
L67				X		X	Plan Fix	The Base of an LDT (Local Descriptor Table) Register May be Non-zero on a Processor Supporting Intel® Extended Memory 64 Technology (Intel® EM64T)
L68				X		X	No Fix	REP STOS/MOVS Instructions with RCX >=2^32 May Cause a System Hang
L69				X		X	Plan Fix	An REP MOVS or an REP STOS Instruction with RCX >= 2^32 May Fail to Execute to Completion or May Write to Incorrect Memory Locations on Processors Supporting Intel® Extended Memory 64 Technology (Intel® EM64T)
L70				X		X	Plan Fix	An REP LODSB or an REP LODSD or an REP LODSQ Instruction with RCX >= 2^32 May Cause a System Hang on Processors Supporting Intel® Extended Memory 64 Technology (Intel® EM64T)
L71				X		X	No Fix	A Data Access which Spans Both the Canonical and the Non-Canonical Address Space May Hang the System
L72				X		X	No Fix	A 64-Bit Value of Linear Instruction Pointer (LIP) May be Reported Incorrectly in the Branch Trace Store (BTS) Memory Record or in the Precise Event Based Sampling (PEBS) Memory Record
L73	X	X	X	X	X	X	Plan Fix	It is Possible That Two specific Invalid Opcodes May Cause Unexpected Memory Accesses

NO.	C0	D0	E0	LE0	G1	LG1	Plans	ERRATA
L74	X	X	X	X	X	X	No Fix	The Processor May Issue Multiple Code Fetches to the Same Cacheline for Systems with Slow Memory
L75	X	X	X	X	X	X	No Fix	Writing the Local Vector Table (LVT) when an Interrupt is Pending May Cause an Unexpected Interrupt
L76	X	X	X	X	X	X	No Fix	IRET under Certain Conditions May Cause an Unexpected Alignment Check Exception
L77	X	X	X	X	X	X	No Fix	L2 Cache ECC Machine Check Errors May be erroneously Reported after an Asynchronous RESET# Assertion
L78	X	X	X	X	X	X	No Fix	Using 2M/4M Pages When A20M# Is Asserted May Result in Incorrect Address Translations
L79	X	X	X	X	X	X	No Fix	Writing Shared Unaligned Data that Crosses a Cache Line without Proper Semaphores or Barriers May Expose a Memory Ordering Issue

No.	Plan	Specification Clarifications
L1		THERMTRIP# De-assertion Clarification for the 478-pin Package.
L2		THERMTRIP# De-assertion Clarification for the 775-pin Package.

No.	Plan	Documentation Changes
		There are no Documentation Changes in this Specification Update revision.

§

General Information

This section contains top marking information for the Intel® Celeron® D Processor on 90 nm Process and in the 478-Pin Package and the Intel® Celeron® D Processor on 90 nm Process and in the 775 Land Package

Figure 1. Example of Sample Package Markings

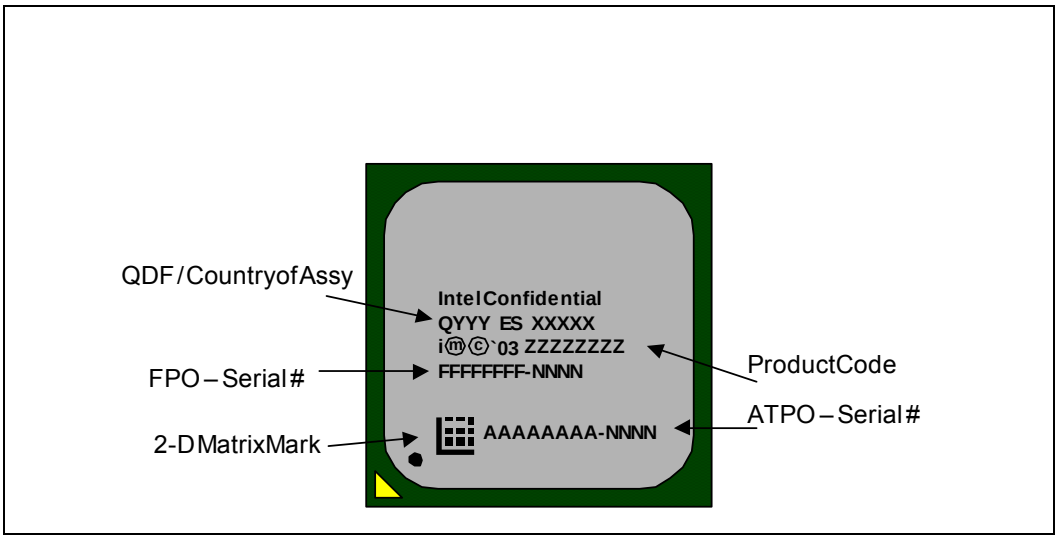


Figure 2. Example of Package Markings

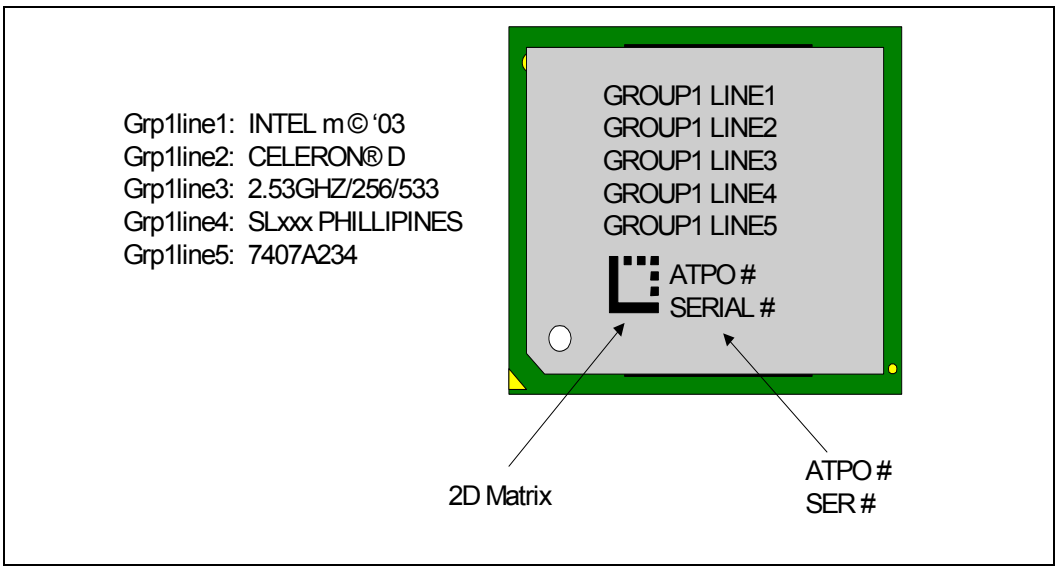
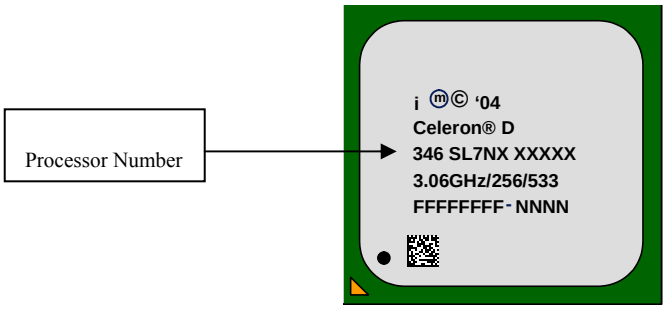


Figure 3. Package Markings with Processor Number



§

Component Identification Information

The Intel® Celeron® D Processor on 90 nm Process and in the 478-Pin Package and the Intel® Celeron® D Processor on 90 nm Process and in the 775 Land Package may be identified by the following values.

Family[1]	Model[2]
1111b	0011b
1111b	0100b

NOTES:

1. The Family corresponds to bits [11:8] of the EDX register after RESET, bits [11:8] of the EAX register after the CPUID instruction is executed with a 1 in the EAX register, and the generation field of the Device ID register accessible through Boundary Scan.
2. The Model corresponds to bits [7:4] of the EDX register after RESET, bits [7:4] of the EAX register after the CPUID instruction is executed with a 1 in the EAX register, and the model field of the Device ID register accessible through Boundary Scan.

§

Table 1. Intel® Celeron® D Processor on 90 nm Process and in the 478-Pin Package and the Intel® Celeron® D Processor on 90 nm Process and in the 775 Land Package Identification Information

S-Spec	Core Stepping	Processor Signature	Processor Number	Core Freq (GHz)	Data Bus Freq (MHz)	L2 Cache Size	Processor Package Revision	Package And Revision	Notes
SL7WS	D0	0F34h	315	2.26	533	256KB	02	478-pin micro-PGA with 35.0 x 35.0 mm FC-mPGA4 package	1
SL7XY	E0	0F41h	315	2.26	533	256KB	02	478-pin micro-PGA with 35.0 x 35.0 mm FC-mPGA4 package	1
SL7C4	C0	0F33h	320	2.40	533	256KB	02	478-pin micro-PGA with 35.0 x 35.0 mm FC-mPGA4 package	1
SL7JV	D0	0F34h	320	2.40	533	256KB	02	478-pin micro-PGA with 35.0 x 35.0 mm FC-mPGA4 package	1
SL7C5	C0	0F33h	325	2.53	533	256KB	02	478-pin micro-PGA with 35.0 x 35.0 mm FC-mPGA4 package	1
SL7ND	D0	0F34h	325	2.53	533	256KB	02	478-pin micro-PGA with 35.0 x 35.0 mm FC-mPGA4 package	1
SL7VQ	E0	0F41h	320	2.40	533	256KB	01	775-land FC-LGA4 37.5 x 37.5 mm	1
SL7TL	E0	0F41h	325J	2.53	533	256KB	01	775-land FC-LGA4 37.5 x 37.5 mm	1
SL7VR	E0	0F41h	325J	2.53	533	256KB	01	775-land FC-LGA4 37.5 x 37.5 mm	1
SL7C6	C0	0F33h	330	2.66	533	256KB	02	478-pin micro-PGA with 35.0 x 35.0 mm FC-mPGA4 package	1
SL7DL	D0	0F34h	330	2.66	533	256KB	02	478-pin micro-PGA with 35.0 x 35.0 mm FC-mPGA4 package	1
SL7TM	E0	0F41h	330J	2.66	533	256KB	01	775-land FC-LGA4 37.5 x 37.5 mm	1
SL7VS	E0	0F41h	330J	2.66	533	256KB	01	775-land FC-LGA4 37.5 x 37.5 mm	1
SL7C7	C0	0F33h	335	2.80	533	256KB	02	478-pin micro-PGA with 35.0 x 35.0 mm FC-mPGA4 package	1
SL7DM	D0	0F34h	335	2.80	533	256KB	02	478-pin micro-PGA with 35.0 x 35.0 mm FC-mPGA4 package	1

Table 1. Intel® Celeron® D Processor on 90 nm Process and in the 478-Pin Package and the Intel® Celeron® D Processor on 90 nm Process and in the 775 Land Package Identification Information

S-Spec	Core Stepping	Processor Signature	Processor Number	Core Freq (GHz)	Data Bus Freq (MHz)	L2 Cache Size	Processor Package Revision	Package And Revision	Notes
SL7TN	E0	0F41h	335J	2.80	533	256KB	01	775-land FC-LGA4 37.5 x 37.5 mm	1
SL7VT	E0	0F41h	335J	2.80	533	256KB	01	775-land FC-LGA4 37.5 x 37.5 mm	1
SL7L2	E0	0F41h	335	2.80	533	256KB	02	478-pin micro-PGA with 35.0 x 35.0 mm FC-mPGA4 package	1
SL7Q9	D0	0F34h	340	2.93	533	256KB	02	478-pin micro-PGA with 35.0 x 35.0 mm FC-mPGA4 package	1
SL7TP	E0	0F41h	340J	2.93	533	256KB	01	775-land FC-LGA4 37.5 x 37.5 mm	1
SL7VV	E0	0F41h	345J	3.06	533	256KB	01	775-land FC-LGA4 37.5 x 37.5 mm	2
SL7TQ	E0	0F41h	345J	3.06	533	256KB	01	775-land FC-LGA4 37.5 x 37.5 mm	1
SL8AW	E0	0F41h	315	2.26	533	256KB	02	478-pin micro-PGA with 35.0 x 35.0 mm FC-mPGA4 package	1
SL7TG	C0	0F33h	325	2.53	533	256KB	02	478-pin micro-PGA with 35.0 x 35.0 mm FC-mPGA4 package	1
SL7TH	C0	0F33h	330	2.66	533	256KB	02	478-pin micro-PGA with 35.0 x 35.0 mm FC-mPGA4 package	1
SL7TJ	C0	0F33h	335	2.80	533	256KB	02	478-pin micro-PGA with 35.0 x 35.0 mm FC-mPGA4 package	1
SL7KX	D0	0F34h	320	2.40	533	256KB	02	478-pin micro-PGA with 35.0 x 35.0 mm FC-mPGA4 package	1
SL7KY	D0	0F34h	325	2.53	533	256KB	02	478-pin micro-PGA with 35.0 x 35.0 mm FC-mPGA4 package	1
SL7VW	E0	0F41h	320	2.40	533	256KB	02	478-pin micro-PGA with 35.0 x 35.0 mm FC-mPGA4 package	1
SL7SS	D0	0F34h	325	2.53	533	256KB	02	775-land FC-LGA4 37.5 x 37.5 mm	1
SL7VX	E0	0F41h	325	2.53	533	256KB	02	478-pin micro-PGA with 35.0 x 35.0 mm FC-mPGA4 package	1

Table 1. Intel® Celeron® D Processor on 90 nm Process and in the 478-Pin Package and the Intel® Celeron® D Processor on 90 nm Process and in the 775 Land Package Identification Information

S-Spec	Core Stepping	Processor Signature	Processor Number	Core Freq (GHz)	Data Bus Freq (MHz)	L2 Cache Size	Processor Package Revision	Package And Revision	Notes
SL7ST	D0	0F34h	330	2.66	533	256KB	02	775-land FC-LGA4 37.5 x 37.5 mm	1
SL7VY	E0	0F41h	330	2.66	533	256KB	02	478-pin micro-PGA with 35.0 x 35.0 mm FC-mPGA4 package	1
SL7KZ	E0	0F41h	330	2.66	533	256KB	02	478-pin micro-PGA with 35.0 x 35.0 mm FC-mPGA4 package	1
SL7SU	D0	0F34h	335	2.80	533	256KB	02	775-land FC-LGA4 37.5 x 37.5 mm	1
SL7VZ	E0	0F41h	335	2.80	533	256KB	02	478-pin micro-PGA with 35.0 x 35.0 mm FC-mPGA4 package	1
SL7RN	D0	0F34h	340	2.93	533	256KB	02	478-pin micro-PGA with 35.0 x 35.0 mm FC-mPGA4 package	1
SL7SV	D0	0F34h	340	2.93	533	256KB	02	775-land FC-LGA4 37.5 x 37.5 mm	1
SL7W2	E0	0F41h	340	2.93	533	256KB	02	478-pin micro-PGA with 35.0 x 35.0 mm FC-mPGA4 package	1
SL7W3	E0	0F41h	345	3.06	533	256KB	02	478-pin micro-PGA with 35.0 x 35.0 mm FC-mPGA4 package	1
SL7NX	E0	0F41h	345	3.06	533	256KB	02	478-pin micro-PGA with 35.0 x 35.0 mm FC-mPGA4 package	1
SL7TU	E0	0F41h	326	2.53	533	256KB	01	775-land FC-LGA4 37.5 x 37.5 mm	1,3
SL7TV	E0	0F41h	331	2.66	533	256KB	01	775-land FC-LGA4 37.5 x 37.5 mm	1,3
SL7TW	E0	0F41h	336	2.80	533	256KB	01	775-land FC-LGA4 37.5 x 37.5 mm	1,3
SL7TX	E0	0F41h	341	2.93	533	256KB	01	775-land FC-LGA4 37.5 x 37.5 mm	1,3
SL7TY	E0	0F41h	346	3.06	533	256KB	01	775-land FC-LGA4 37.5 x 37.5 mm	1,3
SL7NY	E0	0F41h	350	3.20	533	256KB	02	478-pin micro-PGA with 35.0 x 35.0 mm FC-mPGA4 package	1

Table 1. Intel® Celeron® D Processor on 90 nm Process and in the 478-Pin Package and the Intel® Celeron® D Processor on 90 nm Process and in the 775 Land Package Identification Information

S-Spec	Core Stepping	Processor Signature	Processor Number	Core Freq (GHz)	Data Bus Freq (MHz)	L2 Cache Size	Processor Package Revision	Package And Revision	Notes
SL7TZ	E0	0F41h	351	3.20	533	256KB	01	775-land FC-LGA4 37.5 x 37.5 mm	1,3
SL8RZ	E0	0F41h	310	2.13	533	256KB	01	478-pin micro-PGA with 35.0 x 35.0 mm FC-mPGA4 package	1
SL8H5	G1	0F49h	326	2.53	533	256KB	01	775-land FC-LGA4 37.5 x 37.5 mm	1, 3
SL8H7	G1	0F49h	331	2.66	533	256KB	01	775-land FC-LGA4 37.5 x 37.5 mm	1, 3
SL8H9	G1	0F49h	336	2.8	533	256KB	01	775-land FC-LGA4 37.5 x 37.5 mm	1, 3
SL8HB	G1	0F49h	341	2.93	533	256KB	01	775-land FC-LGA4 37.5 x 37.5 mm	1, 3
SL8HD	G1	0F49h	346	3.06	533	256KB	01	775-land FC-LGA4 37.5 x 37.5 mm	1, 3
SL8HF	G1	0F49h	351	3.2	533	256KB	01	775-land FC-LGA4 37.5 x 37.5 mm	1, 3
SL8HS	G1	0F49h	355	3.33	533	256KB	01	775-land FC-LGA4 37.5 x 37.5 mm	1, 3

NOTES:
1. The may be a boxed/tray processor with an unattached Fan Heatsink.
2. This is a boxed Intel® Celeron D processor on 90 nm process with an unattached fan heatsink.
3. This is a EM64T[1] capable part.

Errata

L1. Transaction Is Not Retired after BINIT#

Problem: If the first transaction of a locked sequence receives a HITM# and DEFER# during the snoop phase it should be retried and the locked sequence restarted. However, if BINIT# is also asserted during this transaction, the transaction will not be retried.

Implication: When this erratum occurs, locked transactions will not be retried.

Workaround: None identified.

Status: For the steppings affected, see the *Summary Tables of Changes*.

L2. Invalid Opcode 0FFFh Requires a ModRM Byte

Problem: Some invalid opcodes require a ModRM byte and other following bytes, while others do not. The invalid opcode 0FFFh did not require a ModRM in previous generation microprocessors such as Pentium® II or Pentium III processors, but it is required in the Intel® Pentium® 4 processors, and the Prescott processor.

Implication: The use of an invalid opcode 0FFFh without the ModRM byte may result in a page or limit fault on the Prescott processor. When this erratum occurs, locked transactions will not be retried.

Workaround: To avoid this erratum use ModRM byte with invalid 0FFFh opcode.

Status: For the steppings affected, see the *Summary Tables of Changes*.

L3. Processor May Hang Due to Speculative Page Walks to NonExistent System Memory

Problem: A load operation issued speculatively by the processor that misses the Data Translation Lookaside Buffer (DTLB) results in a page walk. A branch instruction older than the load retires so that this load operation is now in the mispredicted branch path. Due to an internal boundary condition, in some instances the load is not canceled before the page walk is issued.

The Page Miss Handler (PMH) starts a speculative page-walk for the Load and issues a cacheable load of the Page Directory Entry (PDE). This PDE load returns data that points to a page table entry in uncacheable (UC) memory. The PMH issues the PTE Load to UC space, which is issued on the Front Side Bus. No response comes back for this load PTE operation since the address is pointing to system memory, which does not exist.

This load to non-existent system memory causes the processor to hang because other bus requests are queued up behind this UC PTE load, which never gets a response. If the load was accessing valid system memory, the speculative page-walk would successfully complete and the processor would continue to make forward progress.

Implication: Processor may hang due to speculative page walks to non-existent system memory.

Workaround: Page directories and page tables in UC memory space must point to system memory that exists.

Status: For the steppings affected, see the *Summary Tables of Changes*.

L4. Memory Type of the Load Lock Different from Its Corresponding Store Unlock

Problem: A use-once protocol is employed to ensure that the processor in a multi-agent system may access data that is loaded into its cache on a Read-for-Ownership operation at least once before it is snooped out by another agent. This protocol is necessary to avoid a multi-agent livelock scenario in which the processor cannot gain ownership of a line and modify it before that data is snooped out by another agent. In the case of this erratum, split load lock instructions incorrectly trigger the use-once protocol. A load lock operation accesses data that splits across a page boundary with both pages of WB memory type. The use-once protocol activates and the memory type for the split halves get forced to UC. Since use-once does not apply to stores, the store unlock instructions go out as WB memory type. The full sequence on the bus is: locked partial read (UC), partial read (UC), partial write (WB), locked partial write (WB). The use-once protocol should not be applied to load locks.

Implication: When this erratum occurs, the memory type of the load lock will be different than the memory type of the store unlock operation. This behavior (Load Locks and Store Unlocks having different memory types) does not however introduce any functional failures such as system hangs or memory corruption.

Workaround: None identified.

Status: For the steppings affected, see the *Summary Tables of Changes*.

L5. Machine Check Architecture Error Reporting and Recovery May Not Work As Expected

Problem: When the processor detects errors it should attempt to report and/or recover from the error. In the situations described below, the processor does not report and/or recover from the error(s) as intended.

When a transaction is deferred during the snoop phase and subsequently receives a Hard Failure response, the transaction should be removed from the bus queue so that the processor may proceed. Instead, the transaction is not properly removed from the bus queue, the bus queue is blocked, and the processor will hang.

When a hardware prefetch results in an uncorrectable tag error in the L2 cache, MC0_STATUS.UNCOR and MC0_STATUS.PCC are set but no Machine Check Exception (MCE) is signaled. No data loss or corruption occurs because the data being prefetched has not been used. If the data location with the uncorrectable tag error is subsequently accessed, an MCE will occur. However, upon this MCE, or any other subsequent MCE, the information for that error will not be logged because MC0_STATUS.UNCOR has already been set and the MCA status registers will not contain information about the error which caused the MCE assertion but instead will contain information about the prefetch error event.

When the reporting of errors is disabled for Machine Check Architecture (MCA) Bank 2 by setting all MC2_CTL register bits to 0, uncorrectable errors should be logged in the IA32_MC2_STATUS register but no machine-check exception should be generated. Uncorrectable loads on bank 2, which would normally be logged in the IA32_MC2_STATUS register, are not logged.

When one half of a 64 byte instruction fetch from the L2 cache has an uncorrectable error and the other 32 byte half of the same fetch from the L2 cache has a correctable error, the processor will attempt to correct the correctable error but cannot proceed due to the uncorrectable error. When this occurs the processor will hang.

When an L1 cache parity error occurs, the cache controller logic should write the physical address of the data memory location that produced that error into the IA32_MC1_ADDR REGISTER (MC1_ADDR). In some instances of a parity error on a load operation that hits the L1 cache, however, the cache controller logic may write the physical address from a subsequent load or store operation into the IA32_MC1_ADDR register.

When an error exists in the tag field of a cache line such that a request for ownership (RFO) issued by the processor hits multiple tag fields in the L2 cache (the correct tag and the tag with the error) and the accessed data also has a correctable error, the processor will correctly log the multiple tag match error but will hang when attempting to execute the machine check exception handler.

If a memory access receives a machine check error on both 64 byte halves of a 128-byte L2 cache sector, the IA32_MC0_STATUS register records this event as multiple errors, i.e., the valid error bit and the overflow error bit are both set indicating that a machine check error occurred while the results of a previous error were in the error-reporting bank. The IA32_MC1_STATUS register should also record this event as multiple errors but instead records this event as only one correctable error.

The overflow bit should be set to indicate when more than one error has occurred. The overflow bit being set indicates that more than one error has occurred. Because of this erratum, if any further errors occur, the MCA overflow bit will not be updated, thereby incorrectly indicating only one error has been received.

If an I/O instruction (IN, INS, REP INS, OUT, OUTS, or REP OUTS) is being executed, and if the data for this instruction becomes corrupted, the processor will signal a Machine Check Exception (MCE). If the instruction is directed at a device that is powered down, the processor may also receive an assertion of SMI#. Since MCEs have higher priority, the processor will call the MCE handler, and the SMI# assertion will remain pending. However, while attempting to execute the first instruction of the MCE handler, the SMI# will be recognized and the processor will attempt to execute the SMM handler. If the SMM handler is successfully completed, it will attempt to restart the I/O instruction, but will not have the correct machine state due to the call to the MCE handler. This can lead to failure of the restart and shutdown of the processor.

If PWRGOOD is de-asserted during a RESET# assertion causing internal glitches, the MCA registers may latch invalid information.

If RESET# is asserted, then de-asserted, and reasserted, before the processor has cleared the MCA registers, then the information in the MCA registers may not be reliable, regardless of the state or state transitions of PWRGOOD.

If MCERR# is asserted by one processor and observed by another processor, the observing processor does not log the assertion of MCERR#. The Machine Check Exception (MCE) handler called upon assertion of MCERR# will not have any way to determine the cause of the MCE.

The Overflow Error bit (bit 62) in the IA32_MC0_STATUS register indicates, when set, that a machine check error occurred while the results of a previous error were still in the error reporting bank (i.e. The Valid bit was set when the new error occurred). If an uncorrectable error is logged in the error-reporting bank and another error occurs, the overflow bit will not be set.

The MCA Error Code field of the IA32_MC0_STATUS register gets written by a different mechanism than the rest of the register. For uncorrectable errors, the other fields in the IA32_MC0_STATUS register are only updated by the first error. Any further errors that are detected will update the MCA Error Code field without updating the rest of the register, thereby leaving the IA32_MC0_STATUS register with stale information.

When a speculative load operation hits the L2 cache and receives a correctable error, the IA32_MC1_Status Register may be updated with incorrect information. The IA32_MC1_Status Register should not be updated for speculative loads.

The processor should only log the address for L1 parity errors in the IA32_MC1_Status register if a valid address is available. If a valid address is not available, the Address Valid bit in the IA32_MC1_Status register should not be set. In instances where an L1 parity error occurs and the address is not available because the linear to physical address translation is not complete or an internal resource conflict has occurred, the Address Valid bit is incorrectly set.

The processor may hang when an instruction code fetch receives a hard failure response from the Front Side Bus. This occurs because the bus control logic does not return data to the core, leaving the processor empty. IA32_MC0_STATUS MSR does indicate that a hard fail response occurred.

The processor may hang when the following events occur and the machine check exception is enabled, CR4.MCE=1. A processor that has it's STPCLK# pin asserted will internally enter the Stop Grant State and finally issue a Stop Grant Acknowledge special cycle to the bus. If an uncorrectable error is generated during the Stop Grant process it is possible for the Stop Grant special cycle to be issued to the bus before the processor vectors to the machine check handler. Once the chipset receives its last Stop Grant special cycle it is allowed to ignore any bus activity from the processors. As a result, processor accesses to the machine check handler may not be acknowledged, resulting in a processor hang.

Implication: The processor is unable to correctly report and/or recover from certain errors.

Workaround: None identified.

Status: For the steppings affected, see the *Summary Tables of Changes*.

L6. Debug Mechanisms May Not Function As Expected

Problem: If the first transaction of a locked sequence receives a HITM# and DEFER# during the snoop phase it should be retried and the locked sequence restarted. However, if BINIT# is also asserted during this transaction, the transaction will not be Certain debug mechanisms may not function as expected on the processor. The cases are as follows:

When the following conditions occur: 1) An FLD instruction signals a stack overflow or underflow, 2) the FLD instruction splits a page-boundary or a 64 byte cache line boundary, 3) the instruction matches a Debug Register on the high page or cache line respectively, and 4) the FLD has a stack fault and a memory fault on a split access, the processor will only signal the stack fault and the debug exception will not be taken.

When a data breakpoint is set on the ninth and/or tenth byte(s) of a floating point store using the Extended Real data type, and an unmasked floating point exception occurs on the store, the break point will not be captured.

When any instruction has multiple debug register matches, and any one of those debug registers is enabled in DR7, all of the matches should be reported in DR6 when the processor goes to the debug handler. This is not true during a REP instruction. As an example, during execution of a REP MOVSW instruction the first iteration a load matches DR0 and DR2 and sets DR6 as FFFF0FF5h. On a subsequent iteration of the instruction, a load matches only DR0. The DR6 register is expected to still contain FFFF0FF5h, but the processor will update DR6 to FFFF0FF1h.

A Data breakpoint that is set on a load to uncacheable memory may be ignored due to an internal segment register access conflict. In this case the system will continue to execute instructions, bypassing the intended breakpoint. Avoiding having instructions that access segment descriptor registers e.g. LGDT, LIDT close to the UC load, and avoiding serialized instructions before the UC load will reduce the occurrence of this erratum.

Implication: Certain debug mechanisms do not function as expected on the processor.

Workaround: None identified.

Status: For the steppings affected, see the *Summary Tables of Changes*.

L7. Cascading of Performance Counters Does Not Work Correctly When Forced Overflow Is Enabled

Problem: The performance counters are organized into pairs. When the CASCADE bit of the Counter Configuration Control Register (CCCR) is set, a counter that overflows will continue to count in the other counter of the pair. The FORCE_OVF bit forces the counters to overflow on every non-zero increment. When the FORCE_OVF bit is set, the counter overflow bit will be set but the counter no longer cascades.

Implication: The performance counters do not cascade when the FORCE_OVF bit is set.

Workaround: None identified.

Status: For the steppings affected, see the *Summary Tables of Changes*.

L8. EMON Event Counting of x87 Loads May Not Work As Expected

Problem: If a performance counter is set to count x87 loads and floating-point exceptions are unmasked, the FPU Operand (Data) Pointer (FDP) may become corrupted.

Implication: When this erratum occurs, FPU Operand (Data) Pointer (FDP) may become corrupted.

Workaround: This erratum will not occur with floating point exceptions masked. If floating-point exceptions are unmasked, then performance counting of x87 loads should be disabled.

Status: For the steppings affected, see the *Summary Tables of Changes*.

L9. System Bus Interrupt Messages without Data and Which Receive a HardFailure Response May Hang the Processor

Problem: When a System Bus agent (processor or chipset) issues an interrupt transaction without data onto the System Bus, and the transaction receives a HardFailure response, a potential processor hang can occur. The processor, which generates an inter-processor interrupt (IPI) that receives HardFailure response, will still log the MCA error event cause as HardFailure, even if the APIC causes a hang. Other processors, which are true targets of the IPI, will also hang on hardfailure-without-data, but will not record an MCA HardFailure event as a cause. If a HardFailure response occurs on a System Bus interrupt message with data, the APIC will complete the operation so as not to hang the processor.

Implication: The processor may hang.

Workaround: None identified.

Status: For the steppings affected, see the *Summary Tables of Changes*.

L10. Processor Flags #PF Instead of #AC on an Unlocked CMPXC8B Instruction

Problem: If a data page fault (#PF) and alignment check fault (#AC) both occur for an unlocked CMPXC8B instruction, then #PF will be flagged.

Implication: Software that depends #AC before #PF will be affected since #PF is flagged in this case.

Workaround: Remove the software's dependency on the fact that #AC has precedence over #PF. Alternately, if the reload is due to a not present page, reload the page in the page fault handler and then restart the faulting instruction.

Status: For the steppings affected, see the *Summary Tables of Changes*.

L11. FSW May Not Be Completely Restored after Page Fault on FRSTOR or FLDENV Instructions

Problem: If the FPU operating environment or FPU state (operating environment and register stack) being loaded by an FLDENV or FRSTOR instruction wraps around a 64-Kbyte or 4-Gbyte boundary and a page fault (#PF) or segment limit fault (#GP or #SS) occurs on the instruction near the wrap boundary, the upper byte of the FPU status word (FSW) might not be restored. If the fault handler does not restart program execution at the faulting instruction, stale data may exist in the FSW.

Implication: When this erratum occurs, stale data will exist in the FSW.

Workaround: Ensure that the FPU operating environment and FPU state do not cross 64-Kbyte or 4-Gbyte boundaries. Alternately, ensure that the page fault handler restarts program execution at the faulting instruction after correcting the paging problem.

Status: For the steppings affected, see the *Summary Tables of Changes*.

L12. Processor Issues Inconsistent Transaction Size Attributes for Locked Operation

Problem: When the processor is in the Page Address Extension (PAE) mode and detects the need to set the Access and/or Dirty bits in the page directory or page table entries, the processor sends an 8 byte load lock onto the System Bus. A subsequent 8 byte store unlock is expected, but instead a 4 byte store unlock occurs. Correct data is provided since only the lower bytes change, however external logic monitoring the data transfer may be expecting an 8-byte store unlock.

Implication: No known commercially available chipsets are affected by this erratum.

Workaround: None identified.

Status: For the steppings affected, see the *Summary Tables of Changes*.

L13. When the Processor Is in the System Management Mode (SMM), Debug Registers May Be Fully Writeable

Problem: When in System Management Mode (SMM), the processor executes code and stores data in the SMRAM space. When the processor is in this mode and writes are made to DR6 and DR7, the processor should block writes to the reserved bit locations. Due to this erratum, the processor may not block these writes. This may result in invalid data in the reserved bit locations.

Implication: Reserved bit locations within DR6 and DR7 may become invalid.

Workaround: Software may perform a read/modify/write when writing to DR6 and DR7 to ensure that the value in the reserved bits are maintained.

Status: For the steppings affected, see the *Summary Tables of Changes*.

L14. The Processor May Issue Front Side Bus Transactions up to 6 Clocks after RESET# is Asserted

Problem: The processor may issue transactions beyond the documented 3 Front Side Bus (FSB) clocks and up to 6 FSB clocks after RESET# is asserted in the case of a warm reset. A warm reset is where the chipset asserts RESET# when the system is running.

Implication: The processor may issue transactions up to 6 FSB clocks after RESET# is asserted.

Workaround: None identified.

Status: For the steppings affected, see the *Summary Tables of Changes*.

L15. Processor May Hang Under Certain Frequencies and 12.5% STPCLK# Duty Cycle

Problem: If a system de-asserts STPCLK# at a 12.5% duty cycle, and the processor is running below 2 GHz, and the processor thermal control circuit (TCC) on-demand clock modulation is active, the processor may hang. This erratum does not occur under the automatic mode of the TCC.

Implication: When this erratum occurs, the processor will hang.

Workaround: If use of the on-demand mode of the processor's TCC is desired in conjunction with STPCLK# modulation, then assure that STPCLK# is not asserted at a 12.5% duty cycle.

Status: For the steppings affected, see the *Summary Tables of Changes*.

L16. System May Hang if a Fatal Cache Error Causes Bus Write Line (BWL) Transaction to Occur to the Same Cache Line Address As an Outstanding Bus Read Line (BRL) or Bus Read-Invalidate Line (BRIL)

Problem: A processor internal cache fatal data ECC error may cause the processor to issue a BWL transaction to the same cache line address as an outstanding BRL or BRIL. As it is not typical behavior for a single processor to have a BWL and a BRL/BRIL concurrently outstanding to the same address, this may represent an unexpected scenario to system logic within the chipset.

Implication: The processor may not be able to fully execute the machine check handler in response to the fatal cache error if system logic does not ensure forward progress on the System Bus under this scenario.

Workaround: System logic should ensure completion of the outstanding transactions. Note that during recovery from a fatal data ECC error, memory image coherency of the BWL with respect to BRL/BRIL transactions is not important. Forward progress is the primary requirement.

Status: For the steppings affected, see the *Summary Tables of Changes*.

L17. A Write to APIC Task Priority Register (TPR) That Lowers Priority May Seem to Have Not Occurred

Problem: Uncacheable stores to the APIC space are handled in a non-synchronous way with respect to the speed at which instructions are retired. If an instruction that masks the interrupt flag e.g. CLI is executed soon after an uncacheable write to the TPR that lowers the APIC priority the interrupt masking operation may take effect before the actual priority has been lowered. This may cause interrupts whose priority is lower than the initial TPR but higher than the final TPR to not be serviced until the interrupt flag is finally cleared e.g. STI. Interrupts will remain pended and are not lost.

Implication: This condition may allow interrupts to be accepted by the processor but may delay their service.

Workaround: This can be avoided by issuing a TPR Read after a TPR Write that lowers the TPR value. This will force the store to the APIC priority resolution logic before any subsequent instructions are executed. No commercial operating system is known to be impacted by this erratum.

Status: For the steppings affected, see the *Summary Tables of Changes*.

L18. At Core-to-bus Ratios of 16:1 and Above Defer Reply Transactions with Non-zero REQb Values May Cause a Front Side Bus Stall

Problem: Certain processors are likely to hang the Front Side Bus (FSB) if the following conditions are met:

1. A Defer Reply transaction has a REQb[2:0] value of either 010b, 011b, 100b, 110b, or 111b, and
2. The operating bus ratio is 16:1 or higher

When these conditions are met, the processor may incorrectly and indefinitely assert a snoop stall for the Defer Reply transaction. Such an event will block further progress on the FSB.

Implication: If this erratum occurs, the system may hang. Intel has not observed this erratum with any commercially available system.

Workaround: None identified.

Status: For the steppings affected, see the *Summary Tables of Changes*.

L19. Parity Error in the L1 Cache May Cause the Processor to Hang

Problem: If a locked operation accesses a line in the L1 cache that has a parity error, it is possible that the processor may hang while trying to evict the line.

Implication: If this erratum occurs, it may result in a system hang. Intel has not observed this erratum with any commercially available software.

Workaround: None identified.

Status: For the steppings affected, see the *Summary Tables of Changes*.

L20. BPM4# Signal Not Being Asserted According to Specification.

Problem: BPM4# signal is not being asserted according to the specification. This may cause incorrect operation of In-Target Debuggers particularly at higher FSB frequencies.

Implication: In-Target Debuggers may not function at higher than 133/533 MHz FSB.

Workaround: One method is to reduce the FSB common clock frequency to 133 MHz or lower. For higher FSB speeds, In-Target Debuggers have a built-in function (test2010) that tells the hardware to ignore BPM4# assertions. This may degrade the debugger performance but will give correct results.

Status: For the steppings affected, see the *Summary Tables of Changes*.

L21. A 16-bit Address Wrap Resulting from a Near Branch (Jump or Call) May Cause an Incorrect Address to Be Reported to the #GP Exception Handler

Problem: If a 16-bit application executes a branch instruction that causes an address wrap to a target address outside of the code segment, the address of the branch instruction should be provided to the general protection exception handler. It is possible that, as a result of this erratum, that the general protection handler may be called with the address of the branch target.

Implication: A 16-bit software environment that is affected by this erratum will see that the address reported by the exception handler points to the target of the branch rather than the address of the branch instruction.

Workaround: None identified.

Status: For the steppings affected, see the *Summary Tables of Changes*.

L22. Locks and SMC Detection May Cause the Processor to Temporarily Hang

Problem: The processor may temporarily hang in an HT Technology enabled system, if one logical processor executes a synchronization loop that includes one or more bus locks and is waiting for release by the other logical processor. If the releasing logical processor is executing instructions that are within the detection range of the self modifying code (SMC) logic, then the processor may be locked in the synchronization loop until the arrival of an interrupt or other event.

Implication: If this erratum occurs in an HT Technology enabled system, the application may temporarily stop making forward progress. Intel has not observed this erratum with any commercially available software.

Workaround: None identified.

Status: For the steppings affected, see the *Summary Tables of Changes* at the beginning of this section.

L23. PWRGOOD and TAP Signals Maximum Input Hysteresis Higher Than Specified

Problem: The maximum input hysteresis for the PWRGOOD and TAP input signals is specified at 350 mV. The actual value could be as high as 800 mV.

Implication: The PWRGOOD and TAP inputs may switch at different levels than previously documented specifications. Intel has not observed any issues in validation or simulation as a result of this erratum.

Workaround: None identified.

Status: For the steppings affected, see the *Summary Tables of Changes* at the beginning of this section.

L24. Some Front Side Bus I/O Specifications Are Not Met

Problem: The following front side bus I/O specifications are not met:

- The $V_{IH(min)}$ for the GTL+ signals is specified as GTLREF + (0.10 * V_{CC}) [V].
- The $V_{IH(min)}$ for the Asynchronous GTL+ signals is specified as Vcc/2 + (0.10 * V_{CC}) [V].
- Common Clock Output Valid Delay (min) is specified as -0.250 ns.
- Common Clock Input Setup Time is specified as 0.700 ns.
- Source Synchronous Input Setup Time to Strobe is specified as 0.150 ns.
- Source Synchronous Input Hold Time to Strobe is specified as 0.150 ns.

Implication: This erratum can cause functional failures depending upon system bus activity. It can manifest itself as data parity, address parity, and/or machine check errors.

Workaround: Due to this erratum, the system should meet the following voltage levels and processor timings:

- The $V_{IH(min)}$ for GTL+ signals is now GTLREF + (0.20 * V_{CC}) [V].
- The $V_{IH(min)}$ for the Asynchronous GTL+ signals is now Vcc/2 + (0.20 * V_{CC}) [V].
- Common Clock Output Valid Delay (min) is now -0.300 ns.
- Common Clock Input Setup Time is now 0.900 ns.
- Source Synchronous Input Setup Time to Strobe is now 0.350 ns.
- Source Synchronous Input Hold Time to Strobe is now 0.350 ns

Status: For the steppings affected, see the *Summary Tables of Changes*.

L25. Incorrect Physical Address Size Returned by CPUID Instruction

Problem: The CPUID instruction Function 80000008H (Extended Address Sizes Function) returns the address sizes supported by the processor in the EAX register. This Function returns an incorrect physical address size value of 40 bits. The correct physical address size is 36 bits.

Implication: Function 80000008H returns an incorrect physical address size value of 40 bits.

Workaround: None identified.

Status: For the steppings affected, see the *Summary Tables of Changes*.

L26. Incorrect Debug Exception (#DB) May Occur When a Data Breakpoint Is Set on an FP Instruction

Problem: The default Microcode Floating Point Event Handler routine executes a series of loads to obtain data about the FP instruction that is causing the FP event. If a data breakpoint is set on the instruction causing the FP event, the load in the microcode routine will trigger the data breakpoint resulting in a Debug Exception.

Implication: An incorrect Debug Exception (#DB) may occur if data breakpoint is placed on an FP instruction. Intel has not observed this erratum with any commercially available software or system.

Workaround: None identified.

Status: For the steppings affected, see the *Summary Tables of Changes*.

L27. xAPIC May Not Report Some Illegal Vector Errors

Problem: The local xAPIC has an Error Status Register, which records all errors. The bit 6 (the Receive Illegal Vector bit) of this register, is set when the local xAPIC detects an illegal vector in a received message. When an illegal vector error is received on the same internal clock that the error status register is being written (due to a previous error), bit 6 does not get set and illegal vector errors are not flagged.

Implication: The xAPIC may not report some Illegal Vector errors when they occur at approximately the same time as other xAPIC errors. The other xAPIC errors will continue to be reported.

Workaround: None identified.

Status: For the steppings affected, see the *Summary Tables of Changes.*

L28. Memory Aliasing of Pages as Uncacheable Memory Type and Write Back (WB) May Hang the System

Problem: When a page is being accessed as either Uncacheable (UC) or Write Combining (WC) and WB, under certain bus and memory timing conditions, the system may loop in a continual sequence of UC fetch, implicit writeback, and Request for Ownership (RFO) retries.

Implication: This erratum has not been observed in any commercially available operating system or application. The aliasing of memory regions, a condition necessary for this erratum to occur, is documented as being unsupported in the *IA-32 Intel® Architecture Software Developer's Manual*, Volume 3, section 10.12.4, Programming the PAT. However, if this erratum occurs the system may hang.

Workaround: The pages should not be mapped as either UC or WC and WB at the same time.

Status: For the steppings affected, see the *Summary Tables of Changes.*

L29. Interactions Between the Instruction Translation Lookaside Buffer (ITLB) and the Instruction Streaming Buffer May Cause Unpredictable Software Behavior

Problem: Complex interactions within the instruction fetch / decode unit may make it possible for the processor to execute instructions from an internal streaming buffer containing stale or incorrect information.

Implication: When this erratum occurs, an incorrect instruction stream may be executed resulting in unpredictable software behavior.

Workaround: It is possible for the BIOS to contain a workaround for this erratum.

Status: For the steppings affected, see the *Summary Tables of Changes.*

L30. Changes to CR3 Register Do Not Fence Pending Instruction Page Walks

Problem: When software writes to the CR3 register, it is expected that all previous/outstanding code, data accesses and page walks are completed using the previous value in CR3 register. Due to this erratum, it is possible that a pending instruction page walk is still in progress, resulting in an access (to the PDE portion of the page table) that may be directed to an incorrect memory address.

Implication: The results of the access to the PDE will not be consumed by the processor so the return of incorrect data is benign. However, the system may hang if the access to the PDE does not complete with data (e.g. infinite number of retries).

Workaround: None identified at this time

Status: For the steppings affected, see the *Summary Tables of Changes*

L31. The State of the Resume Flag (RF Flag) in a Task-State Segment (TSS) May Be Incorrect

Problem: After executing a JMP instruction to the next (or other) task through a hardware task switch, it is possible for the state of the RF flag (in the EFLAGS register image) to be incorrect.

Implication: The RF flag is normally used for code breakpoint management during debug of an application. It is not typically used during normal program execution. Code breakpoints or single step debug behavior in the presence of hardware task switches, therefore, may be unpredictable as a result of this erratum. This erratum has not been observed in commercially available software.

Workaround: It is possible for the BIOS to contain a workaround for this erratum. (BIOS workaround may exist for C step only).

Status: For the steppings affected, see the *Summary Tables of Changes*

L32. Processor Provides a 4-Byte Store Unlock After an 8-Byte Load Lock

Problem: When the processor is in the Page Address Extension (PAE) mode and detects the need to set the Access and/or Dirty bits in the page directory or page table entries, the processor sends an 8 byte load lock onto the system bus. A subsequent 8 byte store unlock is expected, but instead a 4 byte store unlock occurs. Correct data is provided since only the lower bytes change, however external logic monitoring the data transfer may be expecting an 8 byte load lock.

Implication: No known commercially available chipsets are affected by this erratum.

Workaround: None identified at this time

Status: For the steppings affected, see the *Summary Tables of Changes*

L33. Front Side Bus Machine Checks May be Reported as a Result of On-Going Transactions during Warm Reset

Problem: Processor Front Side Bus (FSB) protocol/signal integrity machine checks may be reported if the transactions are initiated or in-progress during a warm reset. A warm reset is where the chipset asserts RESET# when the system is running.

Implication: The processor may log FSB protocol/signal integrity machine checks if transactions are allowed to occur during RESET# assertions.

Workaround: BIOS may clear FSB protocol/signal integrity machine checks for systems/chipsets which do not block new transactions during RESET# assertions.

Status: For the steppings affected, see the *Summary Tables of Changes*.

L34. CPUID Instruction May Report Incorrect L2 Associativity in Leaf 0x80000006

Problem: L2 associativity reported by CPUID with EAX=80000006H instruction may be incorrect.

Implication: Software may see an incorrect L2 associativity when viewed via CPUID with EAX=80000006H, however, when viewed via CPUID with EAX=4H the associativity value is correct.

Workaround: None identified.

Status: For the steppings affected, see the *Summary Tables of Changes*.

L35. Execution of IRET or INTn Instructions May Cause Unexpected System Behavior

Problem: There is a small window of time, requiring alignment of many internal micro architectural events, during which the speculative execution of the IRET or INTn instructions in protected or IA-32e mode may result in unexpected software or system behavior.

Implication: This erratum may result in unexpected instruction execution, events, interrupts or a system hang when the IRET instruction is executed. The execution of the INTn instruction may cause debug breakpoints to be missed.

Workaround: It is possible for the BIOS to contain a workaround for this erratum.

Status: For the steppings affected, see the *Summary Tables of Changes*

L36. The FP_ASSIST EMON Event May Return an Incorrect Count

Problem: The performance monitoring event, FP_ASSIST, may incorrectly calculate the number of events if denormals or SSE loads are encountered.

Implication: When this erratum occurs, the FP_ASSIST event may not calculate the correct number of events. As a result, performance optimization software such as Intel® VTune™ Performance Analyzer may not be able to take advantage of certain scenarios.

Workaround: None identified.

Status: For the steppings affected, see the *Summary Tables of Changes*

L37. Machine Check Exceptions May Not Update Last-Exception Record MSRs (LERs)

Problem: The Last-Exception Record MSRs (LERs) may not get updated when Machine Check Exceptions occur.

Implication: When this erratum occurs, the LER may not contain information relating to the machine check exception. They will contain information relating to the exception prior to the machine check exception.

Workaround: None identified.

Status: For the steppings affected, see the *Summary Tables of Changes*

L38. MOV CR3 Performs Incorrect Reserved Bit Checking When in PAE Paging

Problem: The MOV CR3 instruction should perform reserved bit checking on the upper unimplemented address bits. This checking range should match the address width reported by CPUID instruction 0x8000008. This erratum applies whenever PAE is enabled.

Implication: Software that sets the upper address bits on a MOV CR3 instruction and expects a fault may fail. This erratum has not been observed with commercially available software.

Workaround: None.

Status: For the steppings affected, see the *Summary Tables of Changes*

L39. Stores to Page Tables May Not Be Visible to Pagewalks for Subsequent Loads without Serializing or Invalidating the Page Table Entry

Problem: Under rare timing circumstances, a page table load on behalf of a programmatically younger memory access may not get data from a programmatically older store to the page table entry if there is not a fencing operation or page translation invalidate operation between the store and the younger memory access. Refer to the IA-32 Intel® Architecture Software Developer's Manual for the correct way to update page tables. Software that conforms to the Software Developer's Manual will operate correctly.

Implication: If the guidelines in the Software Developer's Manual are not followed, stale data may be loaded into the processor's Translation Lookaside Buffer (TLB) and used for memory operations. This erratum has not been observed with any commercially available software.

Workaround: The guidelines in the IA-32 Intel® Architecture Software Developer's Manual should be followed.

Status: For the steppings affected, see the *Summary Tables of Changes*

L40. Data Breakpoints on the High Half of a Floating Point Line Split May Not Be Captured

Problem: When a floating point load which splits a 64-byte cache line gets a floating point stack fault, and a data breakpoint register maps to the high line of the floating point load, internal boundary conditions exist that may prevent the data breakpoint from being captured.

Implication: When this erratum occurs, a data breakpoint will not be captured.

Workaround: None identified.

Status: For the steppings affected, see the *Summary Tables of Changes*.

L41. A Split Store Memory Access May Miss a Data Breakpoint

Problem: It is possible for a data breakpoint specified by a linear address to be missed during a split store memory access. The problem can happen with or without paging enabled.

Implication: This erratum may limit the debug capability of a debugger software.

Workaround: None identified.

Status: For the steppings affected, see the *Summary Table of*

L42. EFLAGS.RF May Be Incorrectly Set after an IRET Instruction

Problem: EFLAGS.RF is used to disable code breakpoints. After an IRET instruction, EFLAGS.RF may be incorrectly set or not set depending on its value right before the IRET instruction.

Implication: A code breakpoint may be missed or an additional code breakpoint may be taken on next instruction.

Workaround: It is possible for the BIOS to contain a workaround for this erratum.

Status: For the steppings affected, see the *Summary Tables of Changes*

L43. Read for Ownership and Simultaneous Fetch May Cause the Processor to Hang

Problem: The processor may hang when it attempts to fetch from cache line X and line X+1 simultaneously with a Read for Ownership to cache line X. If the fetch to cache line X+1 occurs within a small window of time, the processor will detect this as self-modifying code and the Read for Ownership will be infinitely recycled.

Implication: If this erratum occurs, the processor may hang.

Workaround: It is possible for the BIOS to contain a workaround for this erratum.

Status: For the steppings affected, see the *Summary Tables of Changes*

L44. Writing the Echo TPR Disable Bit in IA32_MISC_ENABLE May Cause a #GP Fault

Problem: Writing a '1' to the Echo TPR disable bit (bit 23) in IA32_MISC_ENABLE may incorrectly cause a #GP fault.

Implication: A #GP fault may occur if the bit is set to a '1'.

Workaround: It is possible for the BIOS to contain a workaround for this erratum.

Status: For the steppings affected, see the *Summary Tables of Changes*

L45. Cache Lock with Simultaneous Invalidate External Snoop and SMC Check May Cause the Processor to Hang

Problem: Under rare timing conditions, the processor may hang when it attempts to execute a cache lock to a cache line location while simultaneously there is a go to invalidate external snoop of the same cache line location and the processor is checking for Self-Modifying Code (SMC) of an unrelated cache line.

Implication: If this erratum occurs, the processor may hang.

Workaround: It is possible for the BIOS to contain a workaround for this erratum.

Status: For the steppings affected, see the *Summary Tables of Changes*

L46. IRET Instruction Performing Task Switch May Not Serialize the Processor Execution

Problem: When an IRET instruction is executed and the NT (Nested Task) flag in the EFLAGS register is set, then buffered writes may not be drained to memory before the next instruction is fetched and executed.

Implication: Executing an IRET instruction when the NT flag in the EFLAGS register is set may not ensure that all pending memory transactions have completed.

Workaround: It is possible for the BIOS to contain a workaround for this erratum

Status: For the steppings affected, see the *Summary Tables of Changes*

L47. Incorrect Access Controls to MSR_LASTBRANCH_0_FROM_LIP MSR Registers

Problem: When an access is made to the MSR_LASTBRANCH_0_FROM_LIP MSR register, an expected #GP fault may not happen.

Implication: A read of the MSR_LASTBRANCH_0_FROM_LIP MSR register may not cause a #GP fault.

Workaround: It is possible for the BIOS to contain a workaround for this erratum

Status: For the steppings affected, see the *Summary Tables of Changes*

L48. Recursive Page Walks May Cause a System Hang

Problem: A page walk, accessing the same page table entry multiple times but at different levels of the page table, which causes the page table entry to have its Access bit set may result in a system hang.

Implication: When this erratum occurs, the system may experience a hang.

Workaround: It is possible for the BIOS to contain a workaround for this erratum.

Status: For the steppings affected, see the *Summary Tables of Changes.*

L49. When the Execute Disable Bit Function is Enabled a Page-fault in a Mispredicted Branch May Result in a Page-fault Exception

Problem: If a page-fault in a mispredicted branch occurs in the ITLB, it should not be reported by the processor. However, if the execute disable bit function is enabled (IA32_EFER.NXE = 1) and there is a page-fault in a mispredicted branch in the ITLB, a page-fault exception may occur.

Implication: When this erratum occurs, a page-fault exception may occur.

Workaround: It is possible for the BIOS to contain a workaround for this erratum.

Status: For the steppings affected, see the *Summary Table of Changes*

L50. Execute Disable Bit Set with AD Assist May Cause Livelock

Problem: If Execute Disable Bit is set and the resulting page requires the processor to set the A and/or D bit (Access and/or Dirty bit) in the PTE, then the processor may livelock.

Implication: When this erratum occurs, the processor may livelock resulting in a system hang or operating system failure.

Workaround: It is possible for the BIOS to contain a workaround for this erratum.

Status: For the steppings affected, see the *Summary Table of Changes*

L51. The Execute Disable Bit Fault May Be Reported before Other Types of Page Fault When Both Occur

Problem: If the Execute Disable Bit is enabled and both the Execute Disable Bit fault and page faults occur, the Execute Disable Bit fault will be reported prior to other types of page fault being reported.

Implication: No impact to properly written code since both types of faults will be generated but in the opposite order.

Workaround: It is possible for the BIOS to contain a workaround for this erratum.

Status: For the steppings affected, see the *Summary Table of Changes*

L52. Execute Disable Bit Set with CR4.PAE May Cause Livelock

Problem: If the Execute Disable Bit of IA32_MISC_ENABLE along with PAE in Control Register bit 4 is set (IA32_EFER.NXE & CR4.PAE), the processor may livelock.

Implication: When this erratum occurs, the processor may livelock resulting in a system hang or operating system failure.

Workaround: It is possible for the BIOS to contain a workaround for this erratum.

Status: For the steppings affected, see the Summary Table of Changes

L53. Checking of Page Table Base Address May Not Match the Address Bit Width Supported by the Platform

Problem: If the page table base address, included in the page map level-4 table, page-directory pointer table, page-directory table or page table, exceeds the physical address range supported by the platform (e.g. 36-bit) and it is less than the implemented address range (e.g. 40-bit), the processor does not check if the address is invalid.

Implication: If software sets such invalid physical address in those tables, the processor does not generate a page fault (#PF) upon access to that virtual address, and the access results in an incorrect read or write. If BIOS provides only valid physical address ranges to the operating system, this erratum will not occur.

Workaround: BIOS must provide valid physical address ranges to the operating system.

Status: For the steppings affected, see the *Summary Tables of Changes*.

L54. The IA32_MCi_STATUS MSR May Improperly Indicate that Additional MCA Information Has Been Captured

Problem: When a data parity error is detected and the bus queue is busy, the ADDRV and MISCV bits of the IA32_MCi_STATUS register may be asserted even though the contents of the IA32_MCi_ADDR and IA32_MCi_MISC MSRs were not properly captured.

Implication: If this erratum occurs, the MCA information captured in the IA32_MCi_ADDR and IA32_MCi_MISC may not correspond to the reported machine-check error, even though the ADDRV and MISCV are asserted.

Workaround: None identified.

Status: For the steppings affected, see the *Summary Tables of Changes*.

L55. With TF (Trap Flag) Asserted, FP Instruction That Triggers an Unmasked FP Exception May Take Single Step Trap before Retirement of Instruction

Problem: If an FP instruction generates an unmasked exception with the EFLAGS.TF=1, it is possible for external events to occur, including a transition to a lower power state. When resuming from the lower power state, it may be possible to take the single step trap before the execution of the original FP instruction completes.

Implication: A Single Step trap will be taken when not expected.

Workaround: None identified.

Status: For the steppings affected, see the *Summary Tables of Changes*.

L56. MCA Corrected Memory Hierarchy Error Counter May Not Increment Correctly

Problem: An MCA corrected Memory hierarchy error counter can report a maximum of 255 errors. Due to the incorrect increment of the counter, the number of errors reported may be incorrect.

Implication: Due to this erratum, the MCA counter may report incorrect number of soft errors.

Workaround: None Identified.

Status: For the steppings affected, see the *Summary Tables of Changes*.

L57. BTS (Branch Trace Store) and PEBS (Precise Event Based Sampling) May Update Memory outside the BTS/PEBS Buffer

Problem: If the BTS/PEBS buffer is defined such that:

- The difference between BTS/PEBS buffer base and BTS/PEBS absolute maximum is not an integer multiple of the corresponding record sizes
- BTS/PEBS absolute maximum is less than a record size from the end of the virtual address space
- The record that would cross BTS/PEBS absolute maximum will also continue past the end of the virtual address space

A BTS/PEBS record can be written that will wrap at the 4G boundary (IA32) or 2^64 boundary (EM64T mode), and write memory outside of the BTS/PEBS buffer.

Implication: Software that uses BTS/PEBS near the 4G boundary (IA32) or 2^64 boundary (EM64T mode), and defines the buffer such that it does not hold an integer multiple of records can update memory outside the BTS/PEBS buffer.

Workaround: Define BTS/PEBS buffer such that BTS/PEBS absolute maximum minus BTS/PEBS buffer base is integer multiple of the corresponding record sizes as recommended in the IA-32 Intel® Architecture Software Developers Manual, Volume 3.

Status: For the steppings affected, see the *Summary Tables of Changes*.

L58. Memory Ordering Failure May Occur with Snoop Filtering Third Party Agents after Issuing and Completing a BWIL (Bus Write Invalidate Line) or BLW (Bus Locked Write) Transaction

Problem: Under limited circumstances, the processors may, after issuing and completing a BWIL or BLW transaction, retain data from the addressed cache line in shared state even though the specification requires complete invalidation. This data retention may also occur when a BWIL transaction's self-snooping yields HITM snoop results.

Implication: A system may suffer memory ordering failures if its central agent incorporates coherence sequencing which depends on full self-invalidation of the cache line associated with (1) BWIL and BLW transactions, or (2) all HITM snoop results without regard to the transaction type and snoop results' source.

Workaround: 1. The central agent can issue a bus cycle that causes a cache line to be invalidated (Bus Read Invalidate Line (BRIL) or BWIL transaction) in response to a processor-generated BWIL (or BLW) transaction to ensure complete invalidation of the associated cache line. If there are no intervening processor-originated transactions to that cache line, the central agent's invalidating snoop will get a clean snoop result.

Or

2. Snoop filtering central agents can:

a. Not use processor-originated BWIL or BLW transactions to update their snoop filter information, or

b. Update the associated cache line state information to shared state on the originating bus (rather than invalid state) in reaction to a BWIL or BLW.

Status: For the steppings affected, see the *Summary Tables of Changes*.

L59. Control Register 2 (CR2) Can be Updated during a REP MOVS/STOS Instruction with Fast Strings Enabled

Problem: Under limited circumstances while executing a REP MOVS/STOS string instruction, with fast strings enabled, it is possible for the value in CR2 to be changed as a result of an interim paging event, normally invisible to the user. Any higher priority architectural event that arrives and is handled while the interim paging event is occurring may see the modified value of CR2.

Implication: The value in CR2 is correct at the time that an architectural page fault is signaled. Intel has not observed this erratum with any commercially available software.

Workaround: None identified.

Status: For the steppings affected, see the *Summary Tables of Changes*.

L60. TPR (Task Priority Register) Updates during Voltage Transitions of Power Management Events May Cause a System Hang

Problem: Systems with Echo TPR Disable (R/W) bit (bit [23] of IA32_MISC_ENABLE register) set to '0' (default), where xTPR messages are being transmitted on the system bus to the processor, may experience a system hang during voltage transitions caused by the power management events.

Implication: This may cause a system hang during voltage transitions of power management events.

Workaround: It is possible for the BIOS to contain a workaround for this erratum. The BIOS workaround disables the Echo TPR updates on affected steppings.

Status: For the steppings affected, see the *Summary Tables of Changes*.

L61. VERR/VERW Instructions May Cause #GP Fault When Descriptor Is in Non-canonical Space

Problem: If a descriptor referenced by the selector specified for the VERR or VERW instructions is in non-canonical space, it may incorrectly cause a #GP fault on a processor supporting Intel® Extended Memory 64 Technology (Intel® EM64T).

Implication: Operating systems or drivers that reference a selector in non-canonical space may experience an unexpected #GP fault. Intel has not observed this erratum with any commercially available software.

Workaround: It is possible for the BIOS to contain a workaround for this erratum.

Status: For the steppings affected, see the *Summary Tables of Changes*.

L62. The Base of a Null Segment May Be Non-zero on a Processor Supporting Intel® Extended Memory 64 Technology (Intel® EM64T)

Problem: In IA-32e mode of the Intel EM64T processor, the base of a null segment may be non-zero.

Implication: Due to this erratum, Intel EM64T enabled systems may encounter unexpected behavior when accessing memory using the null selector.

Workaround: It is possible for the BIOS to contain a workaround for this erratum.

Status: For the steppings affected, see the *Summary Tables of Changes*.

L63. Upper 32 Bits of FS/GS with Null Base May Not Get Cleared in Virtual-8086 Mode on Processors with Intel® Extended Memory 64 Technology (Intel® EM64T) Enabled

Problem: For processors with Intel EM64T enabled, the upper 32 bits of the FS and GS data segment registers corresponding to a null base may not get cleared when segments are loaded in Virtual-8086 mode.

Implication: This erratum may cause incorrect data to be loaded or stored to memory if FS/GS is not initialized before use in 64-bit mode. Intel has not observed this erratum with any commercially available software.

Workaround: None identified.

Status: For the steppings affected, see the *Summary Tables of Changes*.

L64. Processor May Fault When the Upper 8 Bytes of Segment Selector Is Loaded from a Far Jump through a Call Gate via the Local Descriptor Table

Problem: In IA-32e mode of the Intel EM64T processor, control transfers through a call gate via the Local Descriptor Table (LDT) that uses a 16-byte descriptor, the upper 8-byte access may wrap and access an incorrect descriptor in the LDT. This only occurs on an LDT with a LIMIT>0x10008 with a 16-byte descriptor that has a selector of 0xFFFC.

Implication: In the event this erratum occurs, the upper 8-byte access may wrap and access an incorrect descriptor within the LDT, potentially resulting in a fault or system hang. Intel has not observed this erratum with any commercially available software.

Workaround: None identified.

Status: For the steppings affected, see the *Summary Tables of Changes*.

L65. Loading a Stack Segment with a Selector that References a Non-canonical Address Can Lead to a #SS Fault on a Processor Supporting Intel® Extended Memory 64 Technology (Intel® EM64T)

Problem: When a processor supporting Intel EM64T is in IA-32e mode, loading a stack segment with a selector which references a non-canonical address will result in a #SS fault instead of a #GP fault.

Implication: When this erratum occurs, Intel EM64T enabled systems may encounter unexpected behavior.

Workaround: None identified.

Status: For the steppings affected, see the *Summary Tables of Changes*.

L66. FXRSTOR May Not Restore Non-canonical Effective Addresses on Processors with Intel® Extended Memory 64 Technology (Intel® EM64T) Enabled

Problem: If an x87 data instruction has been executed with a non-canonical effective address, FXSAVE may store that non-canonical FP Data Pointer (FDP) value into the save image. An FXRSTOR instruction executed with 64-bit operand size may signal a General Protection Fault (#GP) if the FDP or FP Instruction Pointer (FIP) is in non-canonical form.

Implication: When this erratum occurs, Intel EM64T enabled systems may encounter an unintended #GP fault.

Workaround: Software should avoid using non-canonical effective addressing in EM64T enabled processors. BIOS can contain a workaround for this erratum removing the unintended #GP fault on FXRSTOR.

Status: For the steppings affected, see the *Summary Tables of Changes*.

L67. The Base of an LDT (Local Descriptor Table) Register May be Non-zero on a Processor Supporting Intel® Extended Memory 64 Technology (Intel® EM64T)

Problem: In IA-32e mode of an Intel EM64T-enabled processor, the base of an LDT register may be non-zero.

Implication: Due to this erratum, Intel EM64T-enabled systems may encounter unexpected behavior when accessing an LDT register using the null selector. There may be no #GP fault in response to this access.

Workaround: None identified.

Status: For the steppings affected, see the *Summary Tables of Changes*.

L68. REP STOS/MOVS Instructions with RCX >=2^32 May Cause a System Hang

Problem: In IA-32e mode using Intel EM64T-enabled processors, executing a repeating string instruction with the iteration count greater than or equal to 2^32 and a pending event may cause the REP STOS/MOVS instruction to live lock and hang.

Implication: When this erratum occurs, the processor may live lock and result in a system hang. Intel has not observed this erratum with any commercially available software.

Workaround: Do not use strings larger than 4 GB.

Status: For the steppings affected, see the *Summary Tables of Changes*.

L69. An REP MOVS or an REP STOS Instruction with RCX >= 2^32 May Fail to Execute to Completion or May Write to Incorrect Memory Locations on Processors Supporting Intel® Extended Memory 64 Technology (Intel® EM64T)

Problem: In IA-32e mode using Intel EM64T-enabled processors, an REP MOVS or an REP STOS instruction executed with the register RCX >= 2^32, may fail to execute to completion or may write data to incorrect memory locations.

Implication: This erratum may cause an incomplete instruction execution or incorrect data in the memory. Intel has not observed this erratum with any commercially available software.

Workaround: It is possible for the BIOS to contain a workaround for this erratum.

Status: For the steppings affected, see the *Summary Tables of Changes*.

L70. An REP LODSB or an REP LODSD or an REP LODSQ Instruction with RCX >= 2^32 May Cause a System Hang on Processors Supporting Intel® Extended Memory 64 Technology (Intel® EM64T)

Problem: In IA-32e mode using Intel EM64T-enabled processors, an REP LODSB or an REP LODSD or an REP LODSQ instruction executed with the register RCX >= 2^32 may fail to complete execution causing a system hang. Additionally, there may be no #GP fault due to the non-canonical address in the RSI register.

Implication: This erratum may cause a system hang on Intel EM64T-enabled platforms. Intel has not observed this erratum with any commercially available software.

Workaround: It is possible for the BIOS to contain a workaround for this erratum.

Status: For the steppings affected, see the *Summary Tables of Changes*.

L71. A Data Access which Spans Both the Canonical and the Non-Canonical Address Space May Hang the System

Problem: If a data access causes a page split across the canonical to non-canonical address space the processor may livelock which in turn would cause a system hang.

Implication: When this erratum occurs, the processor may livelock, resulting in a system hang. Intel has not observed this erratum with any commercially available software.

Workaround: None identified.

Status: For the steppings affected, see the *Summary Tables of Changes*.

L72. A 64-Bit Value of Linear Instruction Pointer (LIP) May be Reported Incorrectly in the Branch Trace Store (BTS) Memory Record or in the Precise Event Based Sampling (PEBS) Memory Record

Problem: On a processor supporting Intel® EM64T,

- If an instruction fetch wraps around the 4G boundary in Compatibility Mode, the 64-bit value of LIP in the BTS memory record will be incorrect (upper 32 bits will be set to FFFFFFFFh when they should be 0).
- If a PEBS event occurs on an instruction whose last byte is at memory location FFFFFFFFh, the 64-bit value of LIP in the PEBS record will be incorrect (upper 32 bits will be set to FFFFFFFFh when they should be 0).

Implication: Intel has not observed this erratum on any commercially available software.

Workaround: None identified.

Status: For the steppings affected, see the *Summary Tables of Changes*.

L73. It is Possible That Two specific Invalid Opcodes May Cause Unexpected Memory Accesses

Problem: A processor is expected to respond with an undefined opcode (#UD) fault when executing either opcode 0F 78 or a Grp 6 Opcode with bits 5:3 of the Mod/RM field set to 6, however the processor may respond instead, with a load to an incorrect address.

Implication: This erratum may cause unpredictable system behavior or system hang.

Workaround: It is possible for the BIOS to contain a workaround for this erratum.

Status: For the steppings affected, see the *Summary Tables of Changes.*

L74. The Processor May Issue Multiple Code Fetches to the Same Cacheline for Systems with Slow Memory

Problem: Systems with long latencies on returning code fetch data from memory e.g. BIOS ROM, may cause the processor to issue multiple fetches to the same cache line, once per each instruction executed.

Implication: This erratum may slow down system boot time. Intel has not observed a failure, as a result of this erratum, in a commercially available system.

Workaround: None identified.

Status: For the steppings affected, see the *Summary Tables of Changes*.

L75. Writing the Local Vector Table (LVT) when an Interrupt is Pending May Cause an Unexpected Interrupt

Problem: If a local interrupt is pending when the LVT entry is written, an interrupt may be taken on the new interrupt vector even if the mask bit is set.

Implication: An interrupt may immediately be generated with the new vector when a LVT entry is written, even if the new LVT entry has the mask bit set. If there is no Interrupt Service Routine (ISR) set up for that vector the system will GP fault. If the ISR does not do an End of Interrupt (EOI) the bit for the vector will be left set in the in-service register and mask all interrupts at the same or lower priority.

Workaround: Any vector programmed into an LVT entry must have an ISR associated with it, even if that vector was programmed as masked. This ISR routine must do an EOI to clear any unexpected interrupts that may occur. The ISR associated with the spurious vector does not generate an EOI, therefore the spurious vector should not be used when writing the LVT.

Status: For the steppings affected, see the *Summary Tables of Changes*.

L76. IRET under Certain Conditions May Cause an Unexpected Alignment Check Exception

Problem: In IA-32e mode, it is possible to get an Alignment Check Exception (AC#) on the IRET instruction even though alignment checks were disabled at the start of the IRET. This can only occur if the IRET instruction is returning from CPL3 code to CPL3 code. IRETs from CPL0/1/2 are not affected. This erratum can occur if the EFLAGS value on the stack has the AC flag set, and the interrupt handler's stack is misaligned. In IA-32e mode, RSP is aligned to a 16-byte boundary before pushing the stack frame.

Implication: In IA-32e mode, under the conditions given above, an IRET can get an AC# even if alignment checks are disabled at the start of the IRET. This erratum can only be observed with a software generated stack frame.

Workaround: Software should not generate misaligned stack frames for use with IRET.

Status: For the steppings affected, see the *Summary Tables of Changes*.

L77. L2 Cache ECC Machine Check Errors May be erroneously Reported after an Asynchronous RESET# Assertion

Problem: Machine check status MSRs may incorrectly report the following L2 Cache ECC machine-check errors when cache transactions are in-flight and RESET# is asserted:

- Instruction Fetch Errors (IA32_MC2_STATUS with MCA error code 153)
- L2 Data Write Errors (IA32_MC1_STATUS with MCA error code 145)

Implication: Uncorrected or corrected L2 ECC machine check errors may be erroneously reported. Intel has not observed this erratum on any commercially available system.

Workaround: When a real run-time L2 Cache ECC Machine Check occurs, a corresponding valid error will normally be logged in the IA32_MC0_STATUS register. BIOS may clear IA32_MC2_STATUS and/or IA32_MC1_STATUS for these specific errors when IA32_MC0_STATUS does not have its VAL flag set.

Status: For the steppings affected, see the *Summary Tables of Changes*.

L78. Using 2M/4M Pages When A20M# Is Asserted May Result in Incorrect Address Translations

Problem: An external A20M# pin if enabled forces address bit 20 to be masked (forced to zero) to emulates real-address mode address wraparound at 1 megabyte. However, if all of the following conditions are met, address bit 20 may not be masked:

- paging is enabled
- a linear address has bit 20 set
- the address references a large page
- A20M# is enabled

Implication: When A20M# is enabled and an address references a large page the resulting translated physical address may be incorrect. This erratum has not been observed with any commercially available operating system.

Workaround: Operating systems should not allow A20M# to be enabled if the masking of address bit 20 could be applied to an address that references a large page. A20M# is normally only used with the first megabyte of memory.

Status: For the steppings affected, see the *Summary Tables of Changes*.

L79. Writing Shared Unaligned Data that Crosses a Cache Line without Proper Semaphores or Barriers May Expose a Memory Ordering Issue

Problem: Software which is written so that multiple agents can modify the same shared unaligned memory location at the same time may experience a memory ordering issue if multiple loads access this shared data shortly thereafter. Exposure to this problem requires the use of a data write which spans a cache line boundary.

Implication: This erratum may cause loads to be observed out of order. Intel has not observed this erratum with any commercially available software or system.

Workaround: Software should ensure at least one of the following is true when modifying shared data by multiple agents:

- The shared data is aligned
- Proper semaphores or barriers are used in order to prevent concurrent data accesses

Status: For the steppings affected, see the *Summary Tables of Changes*.

§

Specification Changes

There are no specification changes in this Specification Update revision.

The Specification changes listed in this section (If any) apply to the following documents:

- *Intel® Celeron® D Processors 340, 335, 330, 325, and 320 - On 90 nm Process in the 478-pin Package* Datasheet
- *Intel® Celeron® D Processor 3xx Sequence – On 90 nm Process in the 775-Land Package* Datasheet

All Specification Changes will be incorporated into a future version of the appropriate Celeron D processor on 90 nm process and in the 478-pin package/775 land package processor documentation.

§

Specification Clarifications

The Specification clarifications listed in this section (If any) apply to the following documents:

- *Intel® Celeron® D Processors 340, 335, 330, 325, and 320 Datasheet - On 90 nm Process in the 478-pin Package* Datasheet
- *Intel® Celeron® D Processor 3xx Sequence – On 90 nm Process in the 775-Land Package* Datasheet

All Specification Clarifications will be incorporated into a future version of the appropriate Celeron D processor on 90 nm process and in the 478-pin package/775 Land Package processor documentation.

L1. THERMTRIP# De-assertion Clarification for the 478-pin Package

The following text is an updated description of THERMTRIP# signal as defined in Table 4-3 (Signal description table) of the *Intel® Celeron® D Processor 3xx Sequence Datasheet*.

Driving of the THERMTRIP# signal is enabled within 10 us of the assertion of PWRGOOD (provided VIDPWRGD, VCCVID, and Vcc are asserted) and is disabled on de-assertion of PWRGOOD (if VIDPWRGD, VCCVID, or Vcc are not valid, THERMTRIP# may also be disabled). Once activated, THERMTRIP# remains latched until PWRGOOD, VIDPWRGD, VCCVID or Vcc is de-asserted. While the de-assertion of the PWRGOOD, VIDPWRGD, VCCVID or Vcc signal will de-assert THERMTRIP#, if the processor's junction temperature remains at or above the trip level, THERMTRIP# will again be asserted within 10 us of the assertion of PWRGOOD (provided VIDPWRGD, VCCVID, and Vcc are asserted).

L2. THERMTRIP# De-assertion Clarification for the 775-Land Package

The following text is an updated description of THERMTRIP# signal as defined in Table 4-3 (Signal description table) of the *Intel® Celeron® D Processor 3xx Sequence Datasheet*.

Driving of the THERMTRIP# signal is enabled within 10 us of the assertion of PWRGOOD (provided VTTPWRGD, VTT, and Vcc are asserted) and is disabled on de-assertion of PWRGOOD (if VTTPWRGD, VTT, or Vcc are not valid, THERMTRIP# may also be disabled). Once activated, THERMTRIP# remains latched until PWRGOOD, VTTPWRGD, VTT or Vcc is de-asserted. While the de-assertion of the PWRGOOD, VTTPWRGD, VTT or Vcc signal will de-assert THERMTRIP#, if the processor's junction temperature remains at or above the trip level, THERMTRIP# will again be asserted within 10 us of the assertion of PWRGOOD (provided VTTPWRGD, VTT, and Vcc are asserted).

§

Documentation Changes

There are no documentation changes in this Specification Update revision.

The Documentation Changes listed (If any) in this section apply to the following documents:

- *Intel® Celeron® D Processors 340, 335, 330, 325, and 320 Datasheet - On 90 nm Process in the 478-pin Package* Datasheet
- *Intel® Celeron® D Processor 3xx Sequence – On 90 nm Process in the 775-Land Package* Datasheet

All Documentation Changes will be incorporated into a future version of the appropriate Celeron D processor on 90 nm process and in the 478-pin package/775 Land processor documentation

Note: Documentation changes for *IA-32 Intel® Architecture Software Developer's Manual* volumes 1, 2A, 2B, 3A, and 3B will be posted in a separate document *IA-32 Intel® Architecture Software Developer's Manual Documentation Changes*. Follow the link below to become familiar with this file.

http://developer.intel.com/design/pentium4/specupdt/252046.htm

§